DILLON & YUDELL LLP
ATTORNEYS AT LAW

USPTO FACSIMILE TRANSMITTAL SHEET

TO: Examiner Thomas H. Stevens		FROM: Matthew W. Baca, Reg. No. 42,277
ORGANIZATION: US Patent and Trademark Office		DATE: September 15, 2005
ART UNIT: 2123	CONFIRMATION NO.:	TOTAL NO. OF PAGES INCLUDING COVER: 16
FAX NUMBER: 571-273-8300		APPLICATION SERIAL NO: 09/751,803
ENCLOSED: Appeal Brief		ATTORNEY DOCKET NO: AUS920000227US1

☒ URGENT □ FOR REVIEW □ PLEASE COMMENT □ PLEASE REPLY □ PLEASE RECYCLE

NOTES/COMMENTS:

This fax from the law firm of Dillon & Yudell LLP contains information that is confidential or privileged, or both. This information is intended only for the use of the individual or entity named on this fax cover letter. Any disclosure, copying, distribution or use of this information by any person other than the intended recipient is prohibited. If you have received this fax in error, please notify us by telephone immediately at 512.343.6116 so that we can arrange for the retrieval of the transmitted documents at no cost to you.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

ATTY. DOCKET NO.: AUS920000227US1

IN RE APPLICATION OF: §
§
WOLFGANG ROESNER ET AL. § EXAMINER: **THOMAS H. STEVENS**
§
SERIAL NO.: **09/751,803** §
§
FILED: **DECEMBER 29, 2000** § ART UNIT: **2123**
§
FOR: **SIGNAL OVERRIDE FOR** §
SIMULATION MODELS §
§
§

APPEAL BRIEF UNDER 37 C.F.R. 1.192

Mail Stop Appeal Briefs - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

Sir:

This Appeal Brief is submitted in support of an Appeal of the Examiner's final rejection of claims 1, 2, 4-6, 8-10, and 12-15. A Notice of Appeal in this case was filed and received by the patent office on July 15, 2005. Please charge the fee of $500.00 due under 37 C.F.R. § 1.17(c), as well as any additional required fees, to IBM Deposit Account No. 09-0447.

Certificate of Transmission/Mailing

I hereby certify that this correspondence is being facsimile transmitted to the USPTO at 703-872-9306 or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450 on the date shown below.

Typed or Printed Name: *Shenise Ramdeen* *Date:* *September 15, 2005* *Signature:* SRamdeen

REAL PARTY IN INTEREST

The real party in interest in the present Application is International Business Machines Corporation, the Assignee of the present application as evidenced by the Assignment set forth at reel 011427, frame 0638 et. seq. of the USPTO assignment records.

RELATED APPEALS AND INTERFERENCES

There are no other appeals or interferences known to Appellant, the Appellant's legal representative, or assignee, which directly affect or would be directly affected by or have a bearing on the Board's decision in the pending appeal.

STATUS OF CLAIMS

Claims 1, 2, 4-6, 8-10, and 12-15 stand finally rejected by the Examiner, as noted in the final Office Action dated March 14, 2005. The rejection of Claims 1, 2, 4-6, 8-10, and 12-15 is appealed.

STATUS OF AMENDMENTS

Appellant's Amendment A filed on September 7, 2004 was entered by the Examiner as indicated in the final Office Action. No amendment to the claims was proposed or entered subsequent to the Final Rejection dated March 14, 2005.

SUMMARY OF THE CLAIMED SUBJECT MATTER

Appellant's invention may be implemented as a method, system, or a computer program product operable in a computer-aided design and verification system for replaceably overriding a simulated signal originating from a hardware description language (HDL) simulation model. The manner in which this override is implemented leverages the use of non-conventional comment syntax to maintain independence between instrumentation entities and design entities. To this end, independent Claims 1, 5, and 9 recite a method, system, and computer program product for facilitating a signal override in an HDL simulation model in a manner that prevents the instrumentation entity that generates the replacement override signal, from being compiled by the HDL compiler into the digital circuit logic being tested.

As explained in Appellants' Background and on page 91, lines 4-17, overriding signals, in the sense of replacing an original signal with another signal or a processed variation of itself, is particularly useful during simulation testing of HDL models. Important to Appellants' proposed method and system for providing such simulation testing signal override capacity is the nature and function of "instrumentation entities." As explained in the figures, and particularly with reference to **FIGS. 4B-4D** and **13A-13D**, Appellants' proposed invention is directed to leveraging comment syntax such that the HDL compiler can distinguish between "design entities" (i.e. HDL entities included in the digital circuit or system that is the object of simulation testing) and "instrumentation entities" (i.e. HDL entities utilized to perform facilitate the testing of the entities incorporated in the digital circuit design).

Specifically, Appellant's Claim 1 provides a method for "facilitating signal override in a simulation model of a digital circuit design that includes one or more design entities described utilizing a hardware description language (HDL)," comprising "instantiating an instrumentation entity within at least one of said one or more design entities using port mapping fields designated by a non-conventional HDL comment syntax, said non-conventional HDL comment syntax processed by a post-compiler instrumentation load tool to instantiate said instrumentation entity during a post-compile model build process, wherein said non-conventional HDL comment syntax is recognized by an HDL compiler such that the HDL compiler does not instantiate said instrumentation entity into the digital circuit design" (*see specification* page 16, line 34 – page 17, line 16 (referring to **FIG. 3A**), generally describing a design entity **300** as a basic building block in an HDL circuit design model; page 28, lines 19 through page 29 line 3, generally describing "instrumentation entities"; page 31, line 24 – page 32, line 17, with reference to **FIG. 4B** describing an instrumentation entity FXUCHK **410** and **411** instantiated within hierarchically instantiated design entities FXU **321a** and **321b**; page 34, line 13 – page 35, line 17, describing input port mapping fields **453** having specialized comment syntax (--!!) pre-pended thereto (**FIG. 4B**); page 41, line 13 – page 43, line 15, describing, with reference to **FIG. 4D**, the process by which instrumentation load tool **464** instantiates instrumentation entities FPUCHK and FXUCHK (instance data structures **482** shown in **FIG. 4E**) into simulation model **480** following model compilation by HDL compiler **462**; page 41, lines 1-11, with reference to **FIG. 4D** explaining that HDL compiler **462** recognizes instrumentation comment syntax such that

HDL compiler **462** does not instantiate (i.e. generate instance data structures) the instrumentation into the actual design.

Continuing with Claim 1, the signal override capability is enabled by the aforementioned port mapping fields which themselves comprise:

"an input port mapping field containing data representing an input port at which said instrumentation entity receives a designated signal to be overridden;" (see *see specification* page 94, lines 7 – page 95, line 2, with reference to **FIG. 13C**, describing an HDL source code file **1340** in which an input port map comment **1360** connects a signal input **1320** to an instrumentation entity **1306**); and

"an output port mapping field comprising an override signal field specifying the name of said override signal, a port name field specifying the name of the port from which said override signal is output from said instrumentation entity, and a target signal field specifying the name of said designated signal to be overridden" (*see specification* page 95, lines 10 – 25, referring to **FIG. 13C**, describing output declaration comments **1361** having the representative data field form: --!! *<name>* : *out_port* => *target_signal* [*ctrl_port*], where *name* specifies the name associated with the specific override signal, *out_port* specifies the name of the output port providing the override value, and *target_signal* specifies the name of the signal to be overridden).

The invention recited in Claim 5 provides a system for "facilitating signal override in a simulation model of a digital circuit design that includes one or more design entities described utilizing a hardware description language (HDL)," comprising "processing means for instantiating an instrumentation entity within at least one of said one or more design entities using port mapping fields designated by a non-conventional HDL comment syntax, said non-conventional HDL comment syntax processed by a post-compiler instrumentation load tool to instantiate said instrumentation entity during a post-compile model build process, wherein said non-conventional HDL comment syntax is recognized by an HDL compiler such that the HDL compiler does not instantiate said instrumentation entity into the digital circuit design" (*see specification* page 16, line 34 – page 17, line 16 (referring to **FIG. 3A**), generally describing a design entity **300** as a basic building block in an HDL circuit design model; page 28, lines 19 through page 29 line 3, generally describing "instrumentation entities"; page 31, line 24 – page 32, line 17, with reference to **FIG. 4B** describing an instrumentation entity FXUCHK **410** and

411 instantiated within hierarchically instantiated design entities FXU **321a** and **321b**; page 34, line 13 – page 35, line 17, describing input port mapping fields **453** having specialized comment syntax (--!!) pre-pended thereto (**FIG. 4B**); page 41, line 13 – page 43, line 15, describing, with reference to **FIG. 4D**, the process by which instrumentation load tool **464** instantiates instrumentation entities FPUCHK and FXUCHK (instance data structures **482** shown in **FIG. 4E**) into simulation model **480** following model compilation by HDL compiler **462**; page 41, lines 1-11, with reference to **FIG. 4D** explaining that HDL compiler **462** recognizes instrumentation comment syntax such that HDL compiler **462** does not instantiate (i.e. generate instance data structures) the instrumentation into the actual design.

Continuing with Claim 5, the signal override capability is enabled by the aforementioned port mapping fields which themselves comprise:

"an input port mapping field containing data representing an input port at which said instrumentation entity receives a designated signal to be overridden;" (see *see specification* page 94, lines 7 – page 95, line 2, with reference to **FIG. 13C**, describing an HDL source code file **1340** in which an input port map comment **1360** connects a signal input **1320** to an instrumentation entity **1306**); and

"an output port mapping field comprising an override signal field specifying the name of said override signal, a port name field specifying the name of the port from which said override signal is output from said instrumentation entity, and a target signal field specifying the name of said designated signal to be overridden" (*see specification* page 95, lines 10 – 25, referring to **FIG. 13C**, describing output declaration comments **1361** having the representative data field form: --!! *<name>* : *out_port* => *target_signal* [*ctrl_port*], where *name* specifies the name associated with the specific override signal, *out_port* specifies the name of the output port providing the override value, and *target_signal* specifies the name of the signal to be overridden).

The invention recited in Claim 9 provides a computer program product for "facilitating signal override in a simulation model of a digital circuit design that includes one or more design entities described utilizing a hardware description language (HDL)," comprising "processing means for instantiating an instrumentation entity within at least one of said one or more design entities using port mapping fields designated by a non-conventional HDL comment syntax, said non-conventional HDL comment syntax processed by a post-compiler instrumentation load tool to instantiate said instrumentation entity during a post-compile model build process, wherein said non-conventional HDL comment syntax is recognized by an HDL compiler such that the HDL compiler does not instantiate said instrumentation entity into the digital circuit design" (*see specification* page 16, line 34 – page 17, line 16 (referring to **FIG. 3A**), generally describing a design entity **300** as a basic building block in an HDL circuit design model; page 28, lines 19 through page 29 line 3, generally describing "instrumentation entities"; page 31, line 24 – page 32, line 17, with reference to **FIG. 4B** describing an instrumentation entity FXUCHK **410** and **411** instantiated within hierarchically instantiated design entities FXU **321a** and **321b**; page 34, line 13 – page 35, line 17, describing input port mapping fields **453** having specialized comment syntax (--!!) pre-pended thereto (**FIG. 4B**); page 41, line 13 – page 43, line 15, describing, with reference to **FIG. 4D**, the process by which instrumentation load tool **464** instantiates instrumentation entities FPUCHK and FXUCHK (instance data structures **482** shown in **FIG. 4E**) into simulation model **480** following model compilation by HDL compiler **462**; page 41, lines 1-11, with reference to **FIG. 4D** explaining that HDL compiler **462** recognizes instrumentation comment syntax such that HDL compiler **462** does not instantiate (i.e. generate instance data structures) the instrumentation into the actual design.

Continuing with Claim 9, the signal override capability is enabled by the aforementioned port mapping fields which themselves comprise:

"an input port mapping field containing data representing an input port at which said instrumentation entity receives a designated signal to be overridden;" (see *see specification* page 94, lines 7 – page 95, line 2, with reference to **FIG. 13C**, describing an HDL source code file **1340** in which an input port map comment **1360** connects a signal input **1320** to an instrumentation entity **1306**); and

"an output port mapping field comprising an override signal field specifying the name of said override signal, a port name field specifying the name of the port from which said override signal is output from said instrumentation entity, and a target signal field specifying the name of said designated signal to be overridden" (*see specification* page 95, lines 10 – 25, referring to **FIG. 13C**, describing output declaration comments **1361** having the representative data field form: --!! *<name>* : *out_port* => *target_signal* [*ctrl_port*], where *name* specifies the name associated with the specific override signal, *out_port* specifies the name of the output port providing the override value, and *target_signal* specifies the name of the signal to be overridden).

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The rejection of Claims 1, 2, 4-6, 8-10, and 12-15 under 35 U.S.C. §103(a) as being unpatentable over U.S. Pat. No. 5,805792, issued to *Swaboda et al.* (hereinafter *Swaboda*) in view of U.S. Pat. No. 5,937,190, issued to *Gregory et al.* (hereinafter *Gregory*) is to be reviewed on Appeal.

ARGUMENT

The rejection of Claims 1, 5, and 9 under 35 U.S.C. §103(a) as being unpatentable over *Swaboda* and *Gregory* is not well founded and should be reversed.

A. The combination of *Swaboda* and *Gregory* does not disclose each claimed feature of Claims 1, 5, and 9

Regarding independent Claim 1, representative also of independent Claims 5 and 9, the combination of *Swaboda* and *Gregory* fail to disclose or suggest a method for facilitating signal override in an HDL simulation model environment that includes, "instantiating an instrumentation entity within at least one of said one or more design entities using port mapping fields designated by a non-conventional HDL comment syntax, said non-conventional HDL comment syntax processed by a post-compiler instrumentation load tool to instantiate said instrumentation entity during a post-compile model build process, wherein said non-conventional HDL comment syntax is recognized by an HDL compiler such that the HDL compiler does not instantiate said instrumentation entity into the digital circuit design..."

On page 4, the final Office Action incorrectly asserts that at col. 34, lines 48-57, referencing FIG. 31, *Gregory* discloses instantiating an instrumentation entity within at least one of said one or more design entities using port mapping fields designated by a non-conventional HDL comment syntax. Instead, FIG. 31 and col. 34 illustrate and describe a technique in an HDL simulation environment by which a specialized "Synopsys probe_statement" is used to preserve, in logic, a designated portion of unsynthesized HDL lines after synthesis (see col. 34, lines 9-17 with reference to FIG. 31). *Gregory's* "comments" are therefore used to label regions of HDL for preservation after synthesis and are not used to instantiate an instrumentation entity or any other kind of HDL entity.

With continued reference to the grounds for rejecting Claim 1, Appellants disagree with the assertion on page 4 of the final Office Action that post-compiler processing of non-conventional HDL comments is disclosed by *Gregory* at col. 5, line 57 – col. 6, line 36. In fact, this passage explains various aspects of the model synthesis process including pre-code designer specified constraints, writing HDL step and intermediate data structure processing, and does not appear related to post-compile instrumentation of a model.

In further regard to the grounds for rejecting Claim 1, page 5 of the Office Action incorrectly asserts that *Swaboda* discloses that the port mapping fields further include, in part, "an output port mapping field comprising an override signal field specifying the name of said override signal (Swaboda column 26, lines 13-20), a port name field specifying the name of the port from which said override signal is output from said instrumentation entity, and a target signal field specifying the name of said designated signal to be overridden." Col. 26, lines 13-20 explains:

> The AND gate 1543 has two inputs connected to SWINPROG and nclksel. In this way, LOCK register 1351 bits override any other signals when the bits call for locking the domain (test clock JCLK only). However, if register 1351 unlocks any one or more domains (calling for functional clock FCLK to each such domain), each such domain can be locked internally by either the delayed locking MSB output of SRL 1455 or a conjunction of switch in program SWINPROG and clock select nclksel high.

Nowhere in this passage or elsewhere does the combination of *Gregory* and *Swaboda* disclose port mapping having an output port mapping field comprising an override signal field, a port name field, and a target signal field as recited above.

B. There is no motivation or suggestion in *Swaboda* and/or *Gregory* to combine the HDL simulation techniques and systems described by *Gregory* with the non-HDL environment disclosed by *Swaboda*

Appellants disagree with the contention on page 4 of the final Office Action that at the time of the invention it would have been obvious to one of ordinary skill in the art to modify *Gregory* by way of *Swaboda* since it would be advantageous to annotate specific function within the model to aid in the debugging process. Appellants contend that it is manifestly apparent that *Swaboda* is not properly combinable with *Gregory* for the purpose of supporting the rejection of Claim 1, also representing independent Claims 5 and 9. Similar to Appellants' proposed invention, the architecture and methods disclosed by *Gregory* relate entirely to computer-aided circuit analysis environment such as those utilizing HDLs. In such computer-aided simulation environments, simulation programming languages such as HDLs are utilized to generate "circuits" which are really compiled simulation logic. In stark contrast, *Swaboda* utilizes actual circuits for simulation purposes (see abstract and throughout the specification) and is therefore clearly incompatible with *Gregory* at least with respect to providing signal port mapping and override functionality.

B. The rejection of Claims 2, 4, 6, 8, 10, and 12-15 under 35 U.S.C. §103(a) as being unpatentable over *Gregory* and *Swaboda* is not well founded and should be reversed.

Appellants do not concede than the present combination of *Gregory* and *Swaboda* actually teaches or suggests any of the features of these dependent claims; however, these claims are directly or indirectly dependent on the independent claims 1, 5, and 9 which, as contended above by Appellants, have been incorrectly rejected under the references. By extension, the rejections of claims 2, 4, 6, 8, 10, and 12-15 are not well founded and should be reversed.

CONCLUSION

Appellant has pointed out with specificity the manifest error in the Examiner's rejections, and the claim language that renders the invention patentable over the combinations of references. Appellant, therefore, respectfully requests that this case be remanded to the Examiner with instructions to issue a Notice of Allowance for all pending claims.

Respectfully submitted,

Matthew W. Baca
Reg. No. 42,277
DILLON & YUDELL LLP
8911 N. Capital of Texas Highway
Suite 2110
Austin, Texas 78759
512-343-6116

ATTORNEY FOR APPELLANTS

APPENDIX

1. In a computer-aided design and verification system, a method for facilitating signal override in a simulation model of a digital circuit design that includes one or more design entities described utilizing a hardware description language (HDL), said method comprising:

instantiating an instrumentation entity within at least one of said one or more design entities using port mapping fields designated by a non-conventional HDL comment syntax, said non-conventional HDL comment syntax processed by a post-compiler instrumentation load tool to instantiate said instrumentation entity during a post-compile model build process, wherein said non-conventional HDL comment syntax is recognized by an HDL compiler such that the HDL compiler does not instantiate said instrumentation entity into the digital circuit design, said port mapping fields further including:

an input port mapping field containing data representing an input port at which said instrumentation entity receives a designated signal to be overridden; and

an output port mapping field comprising an override signal field specifying the name of said override signal, a port name field specifying the name of the port from which said override signal is output from said instrumentation entity, and a target signal field specifying the name of said designated signal to be overridden.

2. The method of claim 13, wherein said output port mapping field further comprises a control port field specifying an output port for delivering an override enable signal to said signal selection means.

3. (Cancelled)

4. The method of claim 2, further comprising:

instantiating a latch within said simulation model, wherein said latch stores an override disable bit; and

combining said override disable bit with said override enable signal within a logic gate to produce a combined selection signal that determines whether or not said signal selection means overrides said designated signal with said override signal.

5. In a computer-aided design and verification system, a system for facilitating signal override in a simulation model of a digital circuit design that includes one or more design entities described utilizing a hardware description language (HDL), said system comprising:

processing means for instantiating an instrumentation entity within at least one of said one or more design entities using port mapping fields designated by a non-conventional HDL comment syntax, said non-conventional HDL comment syntax processed by a post-compiler instrumentation load tool to instantiate said instrumentation entity during a post-compile model build process, wherein said non-conventional HDL comment syntax is recognized by an HDL compiler such that the HDL compiler does not instantiate said instrumentation entity into the digital circuit design, said port mapping fields further including:

an input port mapping field containing data representing an input port at which said instrumentation entity receives a designated signal to be overridden; and

an output port mapping field comprising an override signal field specifying the name of said override signal, a port name field specifying the name of the port from which said override signal is output from said instrumentation entity, and a target signal field specifying the name of said designated signal to be overridden.

6. The system of claim 14, wherein said output port mapping field further comprises a control port specifying an output port for delivering an override enable signal to said signal selection means.

7. (Cancelled)

8. The system of claim 6, further comprising:

processing means for instantiating a latch within said simulation model, wherein.said latch stores an override disable bit; and

processing means for combining said override disable bit with said override enable signal within a logic gate to produce a combined selection signal that determines whether or not said signal selection means overrides said designated signal with said override signal.

9. In a computer-aided design and verification system, a computer program product for facilitating signal override in a simulation model of a digital circuit design that includes one or more design entities described utilizing a hardware description language (HDL), said computer program product including computer-executable instructions for performing a method comprising:

instantiating an instrumentation entity within at least one of said one or more design entities using port mapping fields designated by a non-conventional HDL comment syntax, said non-conventional HDL comment syntax processed by a post-compiler instrumentation load tool to instantiate said instrumentation entity during a post-compile model build process, wherein said non-conventional HDL comment syntax is recognized by an HDL compiler such that the HDL compiler does not instantiate said instrumentation entity into the digital circuit design, said port mapping fields further including:

an input port mapping field containing data representing an input port at which said instrumentation entity receives a designated signal to be overridden; and

an output port mapping field comprising an override signal field specifying the name of said override signal, a port name field specifying the name of the port from which said override signal is output from said instrumentation entity, and a target signal field specifying the name of said designated signal to be overridden.

10. The computer program product of claim 15, wherein said output port mapping field further comprises a control port field specifying an output port for delivering an override enable signal to said signal selection means.

11. (Cancelled)

12. The computer program product of claim 10, wherein said method further comprises:

instantiating a latch within said simulation model, wherein said latch stores an override disable bit; and

combining said override disable bit with said override enable signal within a logic gate to produce a combined selection signal that determines whether or not said signal selection means overrides said designated signal with said override signal.

13. The method of claim 1, further comprising generating signal selection means within said simulation model for selectively overriding said designated signal with said override signal responsive to said post-compiler instrumentation load tool processing said port mapping fields.

14. The system of claim 5, further comprising processing means for generating signal selection means within said simulation model for selectively overriding said designated signal with said override signal responsive to said post-compiler instrumentation load tool processing said port mapping fields.

15. The computer program product of claim 9, wherein said method further comprises generating signal selection means within said simulation model for selectively overriding said designated signal with said override signal responsive to said post-compiler instrumentation load tool processing said port mapping fields.